

An improved deep bagging convolutional neural network classifier for efficient intrusion detection system

Mathiyalagan Ramasamy¹, Pamela Vinitha Eric²

¹Research Scholar, Department of Computer Science & Engineering, New Horizon College of Engineering, Bengaluru, India

²Department of Computer Science & Engineering, New Horizon College of Engineering, Bengaluru, India

Article Info

Article history:

Received Jun 21, 2021

Revised Sep 1, 2021

Accepted Dec 28, 2021

Keywords:

Deep bagging convolutional
neural network

Dragonfly algorithm

Feature selection

Intrusion detection system

ABSTRACT

In the current trend, the network-based system has substantial jobs, and they have become the targets of attackers. When an intrusion occurs, the security of a computer system is compromised. As a result, we must seek out the best methods for ensuring frameworks. A crucial component of the security management architecture is the intrusion detection system (IDS). To maintain effective network security, the design and implementation of IDS remain an important assessment topic. For intrusion detection, the previous system created an enhanced relevance vector machine (ERVM) classifier. However, intrusion detection is not robust for large-scale intrusion datasets, resulting in a high attack rate. The suggested work developed an improved deep bagging based convolutional neural network (DBCNN) for intrusion detection to address this issue. Preprocessing, feature selection, and classification are three processes included in the proposed framework. The KDD dataset is preprocessed in this stage using the kalman filter method. The feature selection is then carried out using the inertia weight based dragonfly method (IWDA). Finally, the DBCNN classifier successfully identifies interruption assaults. The KDD dataset is used to test the new model. The test results show that the proposed work accomplishes better execution contrasted and the current framework as far as accuracy, precision, recall and f-measure.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mathiyalagan Ramasamy

Research Scholar, Visvesvaraya Technological University

Belagavi, India

E-mail: mathi.prajval@gmail.com

1. INTRODUCTION

Computer networks have been increasingly important in applications. Every day, the number of connected devices for this application grows, generating significant amounts of data to send and process. An attack/intrusion is a type of unauthorised access that aims to obtain access to systems in order to undermine their confidentiality, availability, and integrity. The intrusion detection system is in charge of monitoring hostile activities on the system and issuing alerts if such an assault occurs (IDS). IDS provides protection against attackers [1]. Network-based and host-based attacks are the two types of IDS. Network-based assaults are anomaly-based attacks that are identified through the interconnections of computer systems, and the system can communicate with other systems via routers and switches, as well as send attacks through this. Host-based assaults may be detected on a single computer system, and they are also simple to defend against. These issues are caused by external devices that are connected to the systems. Web-based attacks are also possible while connected to the internet, and the attacks are disseminated to other systems via email and downloads.

Data mining and machine learning techniques are commonly used in this IDS development. The most critical features for the entire network are chosen without losing information, according to network feature selection. To handle the uneven network traffic, study [2] discusses convolutional neural networks (CNN) are a deep learning method based on IDS. In terms of improving IDS performance, existing techniques are still insufficient. In this paper, an improved deep learning-based approach has been utilised to address difficulties with existing systems in terms of IDS performance and generalisation error reduction. The contribution of this paper is as shown in:

- To manage missing values, the input raw IDS data set is preprocessed with an upgraded kalman filter.
- After then, the sent data is employed in the feature selection process. The improved inertia weight dragonfly optimizer is an evolutionary method described in this suggested work for picking important features. The weight and number of iterations will offer the best solution for feature selection. To produce the best feature selection, a dragon fly optimizer with inertia weight adjustment is applied.
- Deep bagging CNN (DBCNN) are used for classification. At contrast to classic CNN, DBCNN uses a bagging operation in the output layer to improve classification accuracy. With ensemble classifiers and maximum voting, this is introduced in the training process. CNN with bagging can reduce generalisation error, training time, and enhance classification performance while reducing noise.

The recommended feature selection-based classification on IDS data has been tested and compared to existing feature selection and classification methods in terms of assessment metrics.

The paper has been coordinated as; section 2 depicts about the audit of the writing, section 3 presents developmental based element choice and profound learning grouping draws near, section 4 examines about the tested outcomes and section 5 finishes up the paper with future bearings.

2. LITERATURE REVIEW

This section discusses the different literature and recent studies on IDS. The NSL-KDD data set is used to test several classification algorithms in paper [3]. Using the WEKA tool, this can investigate protocols with intruder attacks. To boost classification accuracy, they applied CFS-based dimensionality reduction. A paper [4] presented an IDS based on the least square support vector machine (LSSVM). To handle linear and nonlinear correlated features, it used a mutual information-based feature selection method [5]. They used the datasets KDD cup 99, NSL-KDD, and Kyoto 2006+. The proposed method outperformed existing algorithms in terms of accuracy and computing cost. Filter based feature selection algorithm such as information gain, correlation based feature selection, principal component analysis and wrapper based feature selection called genetic algorithm, artificial bee colony and particle swarm optimization for network IDS are discussed in paper [6]. SVM was employed as a classifier, and the NSL-KDD dataset was used. They came to the conclusion that using a wrapper-based feature selection strategy improves classification accuracy. An ensemble technique was presented in paper [7] to improve IDS performance. With a tree-based classifier, they applied two methods: boosting and bagging. For the evaluation, they employed 35 features and the NSL-KDD dataset. They came to the conclusion that bagging using the J48 classifier is more effective [8]-[10].

3. PROPOSED METHODOLOGY

Preprocessing, feature extraction, and classification are three aspects of the proposed method. Figure 1 depicts a high-level overview of the technique. The network data set is initially partitioned into three datasets, such as training and testing, in a 6:4 ratios. Preprocessing is important in all classification algorithm techniques since it improves accuracy. This preprocessing technique removes the irrelevant and missing data from the original data. In this proposed work, first, Kalman filter (KF) based pre-processing is done to handle the missing values and data that are out of range for further processing. This preprocessed data is then given as input to the feature selection phase. The irrelevant features in the dataset will affect the performance of the network traffic classification in terms of accuracy and make the system as slow. Second, IDS based on optimal feature selection using the evolutionary method called improved inertia weight based dragon fly optimizer has been proposed. This is used to remove the irrelevant features and select the relevant features for further processing. Until the stopping condition met, the relevant features are selected using the proposed approach. Selected features are then trained and classified using the deep learning algorithm called DBCNN. These suggested IW-DFO based DBCNN classification approaches were tested against the intrusion detection dataset to demonstrate the efficacy of the presented work in terms of performance metrics.

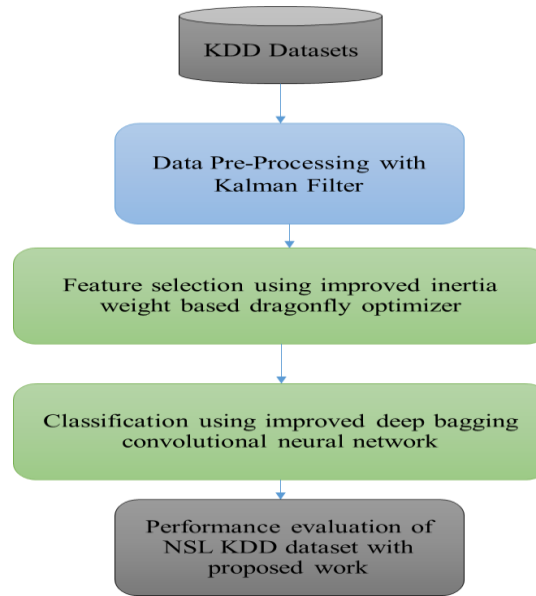


Figure 1. Overview of proposed IDS approach

3.1. Feature selection using proposed improved inertia weight based dragon fly optimizer (IIW-DFO)

The Department of Fisheries and Oceans is replicating dragonfly behaviour for the purpose of migration or hunting. Swarming can take two forms: static and dynamic. Tiny groups of dragonflies hunt other swarms in a small region with local movement of dramatic shifts in a motionless swarm. A big number of dragonflies fly in a single direction over a long distance together in a dynamic swarm [11]. The most significant elements of swarm intelligence techniques are the exploration and utilisation of this static and dynamic activity. The five weights must be modified to maximise the exploration and exploitation process. Separation, alignment, cohesion, food attraction, and DFO's opponent distraction have all been mathematically stated:

- a. Separation: it is the individual avoidance from other neighborhood to separate themselves from other agents. It is calculated as in (1) from [12]

$$S_i = -\sum_{j=1}^N X - X_j \quad (1)$$

where, X =current position, X_j =position of the j^{th} neighbor, N =number of neighbors of the dragonfly, S =separation motion of the i^{th} individual.

- b. Alignment: it is the matching of velocity of the individual to the neighbor individual. It is the agent setting of velocity in terms of velocity vector of the neighbor dragonflies. It is calculated as in the (2)

$$A_i = \frac{\sum_{j=1}^N V_j}{N} \quad (2)$$

Where, A_i =alignment motion of the i^{th} individual and V_j =velocity of the j^{th} neighborhood

- c. Cohesion: it is the measurement of the individual towards the center of the neighborhood. It is represented as in (3)

$$C_i = \frac{\sum_{j=1}^N X_j}{N} - X \quad (3)$$

where, C_i =cohesion of the i^{th} individual, N =size of neighborhood.

- d. Attraction towards food: the dragonfly movement towards the attraction of food is represented in (4).

$$F_i = X^+ - X \quad (4)$$

where F_i =attraction of food of i^{th} individual, X^+ =position of the source of food.

- e. Distraction from enemies: dragonflies stay away from enemies which is represented in (5)

$$E_i = X^- + X \quad (5)$$

where E_i =distraction motion of i th individual enemy, X^- =position of enemy. The position of the individual dragonfly has been updated by considering two factors such as step factor ΔX and position vector X . The step vector is same as in the velocity vector of the PSO algorithm [13] which is defined as [14] in the (6).

$$\Delta X_{t+1} = (S_i + A_i + C_i + F_i + E_i) + \omega \Delta X_t \quad (6)$$

where, ω =inertia weight and t =counter for the iteration. The appropriate selection of this inertia weight with least number of iterations produce the optimal solution. In this proposed work, the inertia weight of the step vector has been improved as in (7)

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{t_{max}} \cdot t \quad (7)$$

where, ω_{max} and ω_{min} =starting and ending values of the dragonfly, t_{max} =maximum iteration and t - number of iterations. The weight and the maximum number of iterations are inversely proportional. Increasing the number of iterations, the weight value gets decreased leads to global search ability as strong. Once the step vector calculation over, the position vector has been updated as in (8).

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (8)$$

Pseudo code of IIW-DFO:

```
Input: Dragonfly population and step vector X i,i=(1,2,...n)
The first step is to iterate as many times as possible (t max)
Step 2: Dragonfly objective values are determined.
Step 3: Update the food source and enemy.
Step 4: Using (1), determine five weight factors such as S, A, C, F, and E. (5)
Step 5: Update the radius of your neighbours
Step 6: if the dragon fly has a neighbour
Step 7: Using the velocity vector [15], update it (6)
Step 8: Using the inertia weight, update the inertia weight (7)
Step 9: Using the position vector, update it (8)
Step 10: if not,
Step 11: levy flight [16] is used to update the position vector.
Step 12: if everything else fails, call it a day.
Step 13: The dragon fly's new position is modified based on the changeable boundaries.
Step 14: come to an end
Output: return selected features set
```

The enhanced inertia weight is applied to the enhanced inertia weight is applied to the five weight vectors, velocity, and position vectors until the maximum iteration is reached. If the dragons fly has any neighbours, the position is updated as well. For intruder detection [17]-[19] the optimised features are now fed into a deep learning-based classification technique called DBCNN.

3.2. Classification using proposed deep bagging convolutional neural network

CNN is differentiated with traditional neural network in terms of the convolution layer and pooling layer which is also responsible for feature extraction. Deep CNN can take input as image or audio or any other format. Those input data are preprocessed to get better result. The convolution layer convolutes the inputted data to select the features objects and transmit the results to sub layer. The network training will train the whole network parameters for this convolution. In this layer, the activation function has been applied for nonlinear activation. Pooling layer is used for sub sampling to reduce the data that are generated by the convolution layer. Full connection layer is for classification which is the global operation. Each node in the full connection layer is connected to all the node of previous layers. The output layer is responsible for to produce the classification result with softmax function. Classification performance of the deep CNN has been improved with the bagging operation that are replaced the output layer of the traditional CNN. The results from the convolution and pooling layer are given as input to the bagging ensemble classifier. Classification output is based on the maximum voting of the ensemble.

The bagging method can introduce the bootstrap sampling into the training process of the network. This will reduce the generalization error, training time, reduced noise and improve the classification accuracy. Structure of the improved deep bagging CNN is shown in Figure 2. Let L be the number of network layers. The size of the convolution kernel is k . the kernel matrix dimension is declared as D . s is the filling size and P represents the convolution kernel moving. Steps involved in the DBCNN are as shown in:

Input: Data set D , number of features n

Step 1: Initialize parameters: Initialize the weight w , bias b and the maximum number iteration T and the threshold for the iteration ϵ .

Step 2: Training phase: It consists of forward, backward propagation, and weight and bias updates.

Step 3: Forward propagation: Training data set are given as input and the output is calculated.

For $cl=2$ to $L-1$

- If cl is the convolution layer, then the missing data after filling (a^{cl}) is represented in (9)

$$a^{cl} = ReLU(z^{cl}) = ReLU(a^{cl} \times w^{cl} + b^{cl}) \quad (9)$$

- If cl is the pool layer then,

$$a^{cl} = pool(a^{cl-1}) \quad (10)$$

- If cl is full connection layer then,

$$a^{cl} = \sigma(z^{cl}) = \sigma(w^{cl}a^{cl-1} + b^{cl}) \quad (11)$$

End for

- The output layer of L is represented in (12)

$$a^L = softmax(z^L) = softmax(w^La^{L-1} + b^L) \quad (12)$$

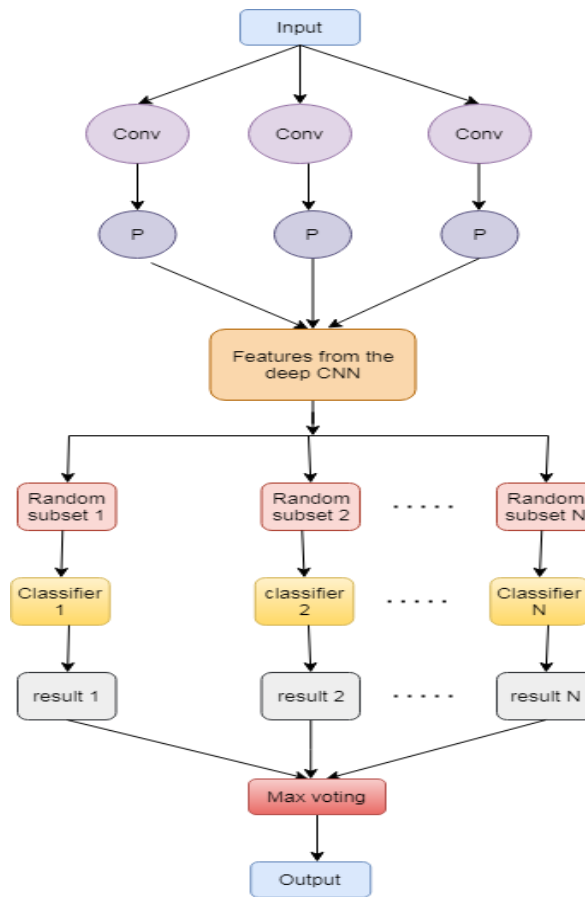


Figure 2. Structure of improved deep bagging CNN (IDBCNN)

Step 4: Backward propagation: This progression used to ascertain the blunder between real yield and relating yield. For $cl=2$ to $L-1$

$$\text{If } cl \text{ is the fully connection layer then, } \delta^{i,cl} = (w^{cl+1})^T \cdot \delta^{i,cl+1} \ominus \sigma(z^{i,cl}) \quad (13)$$

$$\text{If } cl \text{ is the convolution layer then, } \delta^{i,cl} = \delta^{i,cl} \times rot \ 180 \ w^{cl+1} \ominus \sigma(z^{i,cl}) \quad (14)$$

$$\text{If } cl \text{ is pool layer then, } a^{cl} = upsample(\delta^{i,cl+1}) \ominus \sigma(z^{i,cl}) \quad (15)$$

End for

Step 5: weight and bias update: to minimize the error, the weight and bias matrix are updated.

For cl=2 to L-1

– If cl is the fully connection layer then,

$$w^{cl} = w^{cl} - \alpha \sum_{i=1}^m \delta^{i,cl} (a^{i,cl-1})^T \quad (16)$$

$$b^{cl} = b^{cl} - \alpha \sum_{i=1}^m \delta^{i,cl} \quad (17)$$

– If cl is the convolution layer then,

$$w^{cl} = w^{cl} - \alpha \sum_{i=1}^n \delta^{i,cl} \times (a^{i,cl-1}) \quad (18)$$

$$b^{cl} = b^{cl} - \alpha \sum_{i=1}^m \sum_{\mu,v} (\delta^{i,cl})_{\mu,v} \quad (19)$$

End for

Step 6: termination condition checking: *if* ($|a^{t+1} - a^t| < \varepsilon$ or $t < T$) *then loop ends*

Else go to step 1.

Step 7: Bagging: N is the number of base classifiers and the classification label is defined as $Y = \{-1, +1\}$. The bagging method is declared in (20),

$$Y = H(x) = \text{sign}(\sum_{i=1}^N h_i(x)) \quad (20)$$

Step 8: Output: return Y and the relation coefficient matrix w and b.

Hence, the evolutionary based feature selection algorithm called inertia dragonfly optimizer selects the optimal number of relevant features. Deep learning based proposed classification with bagging concept will classify the IDS data with high accuracy with low noise.

4. RESULTS AND DISCUSSION

This section discusses the outcomes of the experiments and the proposed feature selection and classification on IDS. On the NSL-KDD dataset, binary classification was employed, and it was implemented using the keras python deep learning framework.

4.1. Evaluation using performance metrics

The proposed IIWDFO-IDBCNN-IDS system is compared to existing systems in order to evaluate performance utilising performance metrics such as accuracy, FPR, FNR, sensitivity/TPR, specificity/TNR, and recall/attack detection rate (ADR) [20]. The formulae for the evaluation metrics are as:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (21)$$

$$FPR = \frac{FP}{FP+TN} \quad (22)$$

$$FNR = \frac{FN}{FN+TP} \quad (23)$$

$$SN = \frac{TP}{TP+FN} \quad (24)$$

$$SP = \frac{TN}{TN+FP} \quad (26)$$

$$ADR = \frac{TP}{TP+FN} \quad (27)$$

4.2. Performance evaluation based on NSL feature selection approaches

Table 1 shows the performance depending on the original and selected input qualities, IDS performance varies. There are representations of the original 41 features, normalised 41 features, and normalised with feature selection using IIWDFO selected 7 features. Table 1 shows how vital it is to use a two-step normalisation approach to eliminate network traffic data. The feature selection procedure also addresses the issue of overfitting and improves the IDS' overall performance in order to improve classification accuracy. Reduce the rate of error and the time it takes to identify it, as well as the computing complexity.

Table 1. Performance evaluation of the proposed feature selection algorithm

Evaluation metrics	Input features		
	Original features	Normalized features	Selected features by IIW-DFO
No of selected features	41	41	7
Accuracy	92.43	96.37	98.91
FPR	0.016	0.015	0.011
FNR	0.178	0.051	0.018
ADR (%)	89.31	95.31	98.03
Training time (s)	231.39	13.023	3.92
Testing time (s)	272.1	32.31	9.03

The performance of the proposed work's feature selection is compared to those of existing FS on IDS, such as standard DFO [21], FMIFS [22], FLCFS [23], and IGDFOPSOCCNN [3]. Table 2 displays the experimental outcomes.

Table 2. Performance evaluation of proposed IIIW-DFO feature selection with other IDS FS approaches

Metrics	Feature selection approaches				
	DFO	FMIFS	FLCFS	IGDFOPSOCCNN	Proposed IIWDFO
Selected features	23	18	22	16	7
ACC	92.1	88.9	91.23	92.81	98.72
SN	91.92	87.02	90.82	91.02	98.88
SP	87.92	90.82	92.72	88.06	98.02
ADR	93.82	88.61	91.02	90.52	98.34

Table 2 illustrates that, when compared to other existing contemporary techniques, the proposed IIW-DFO FS achieves high accuracy with low complexity analysis, meaning that these selected features would improve classification accuracy and protect the computer network from intruders. The graphical evaluation is depicted in Figure 3. The graphical figure also shows that the proposed FS strategy outperforms existing techniques in terms of accuracy, sensitivity, specificity, and attack detection rate.

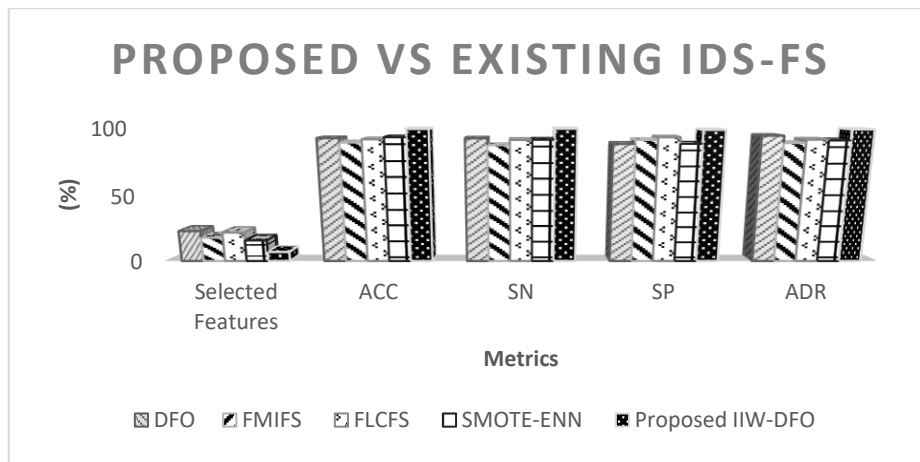


Figure 3. Existing with proposed IIW-DFO

4.4. Performance evaluation of proposed IIWDFO-IBCDNN IDS with existing IDS systems

We compare our proposed convolutional deep neural network based IDS to current IDS systems such as DMNB [24], DBN-SVM [25], PSOM [1], and IGDFOPSOCCNN [3] to prove the deep neural networkbased IDS systems. The evaluation's results are shown in Table 3.

Table 3. Performance evaluation of various existing vs proposed IDS systems

IDS systems	No of Selected Features	Accuracy (%)	FPR
DMNB	41	96.01	1.76
DBN-SVM	41	91.53	2.03
PSOM	10	94.82	3.12
IGDFOPSOCCNN	6	94.02	0.52
Proposed IIWDFO-IDBCNN	7	98.71	0.12

The suggested improved inertia weight dragon fly optimizer with improved DBCNN based IDS has higher accuracy and lower FPR rate than other current IDS techniques, including our earlier work, based on the experimented results. The suggested system detects intruders with an accuracy of 98.71 percent and a false positive rate of 0.12. Because of the optimization procedure, the IIWDFO-IBCDNN IDS achieves such great accuracy. Figure 4 exhibit a graphical representation of these data for a better understanding.

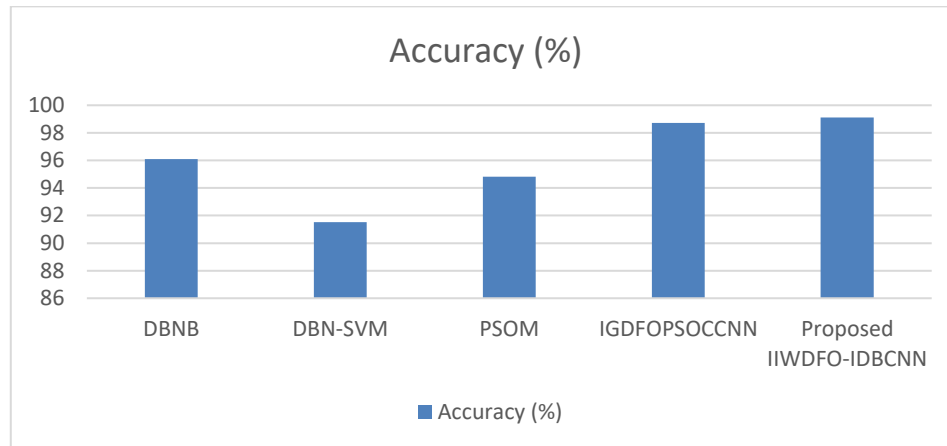


Figure 4. Accuracy comparison of various IDS

As a result, the various IDS process tried results reveal that our proposed IIW-DFO-IDBCDNN has high classification accuracy, low error, and low computing complexity. Even while our previous work is better at detecting intruders, this work using the optimization strategy will increase classification accuracy as well. The IIWDFO-based IDBCDNN beats other existing algorithms in identifying intruders with high accuracy and low error in terms of efficiency and accurate categorization.

5. CONCLUSION

This research presents a feature selection method based on the inertia weight dragonfly optimizer and an upgraded DBCNN for IDS. Due to the enormous number of features and amount of data in the data set, an enhanced proposed classification technology based on evolutionary and deep learning algorithms is provided to increase classification accuracy and forecast network intruders. The easiest way to locate relevant characteristics is to use an optimization-based feature selection method. The accuracy and stability of the system will be increased by a deep convolutional network with bagging. The NSL KDD dataset is used to develop the suggested system. The algorithm's efficiency is demonstrated by a comparison of known contemporary algorithms in terms of feature selection and classification. The results reveal that the suggested evolutionary-based deep learning method outperforms in terms of accuracy (99.11%) and false positive rate (0.8). In the future, the proposed technique will be tested on a small number of datasets with the goal of improving the attack detection rate in the NSL KDD dataset. As a result, the proposed IDS system will enhance classification accuracy while reducing generalisation error, training time, and noise.





REFERENCES

- [1] M. M. Sakr, M. A. Tawfeeq and A. B. El-Sisi, "Filter Versus Wrapper Feature Selection for Network Intrusion Detection System," *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, Cairo, Egypt, 2019, pp. 209-214, doi: 10.1109/ICICIS46948.2019.9014797
- [2] X. Zhang, J. Ran and J. Mi, "An Intrusion Detection System Based on Convolutional Neural Network for Imbalanced Network Traffic," *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, Dalian, China, 2019, pp. 456-460, doi: 10.1109/ICCSNT47585.2019.8962490.
- [3] K. S. Bhuvaneshwari, K. Venkatachalam, S. Hubálovský, P. Trojovský and P. Prabu, "Improved Dragonfly Optimizer for Intrusion Detection Using Deep Clustering CNN-PSO Classifier," *Computers, Materials and Continua*, vol. 77, no. 3, pp.5949-5965 2022, DOI:10.32604/cmc.2022.020769
- [4] S. M. Kasongo and Y. Sun, "A Deep Learning Method with Filter Based Feature Engineering for Wireless Intrusion Detection System," in *IEEE Access*, vol. 7, pp. 38597-38607, 2019, doi: 10.1109/ACCESS.2019.2905633.
- [5] R. Mathiyalagan and P. V. Eri, "An Efficient Intrusion Detection System Using Improved Bias Based Convolutional Neural Network Classifier" in *Turkish Journal of Computer and Mathematics Education*, vol. 12 no.6, 2021, doi: 10.17762/turcomat.v12i6.5689.





- [6] M. A. Ambusaidi, X. He, P. Nanda and Z. Tan, "Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm," in *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986-2998, 1 Oct. 2016, doi: 10.1109/TC.2016.2519914.
- [7] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459-471, 2007, doi: 10.1007/s10898-007-9149-x.
- [8] J. M. Abdullah and R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. ICNN, Piscataway*, Newyork, vol. 4, pp. 1942-1948, 1995.
- [9] T. Mehmod and H. B. M. Rais, "Ant colony optimization and feature selection for intrusion detection," in *Advances in Machine Learning and Signal Processing, New York, NY, USA: Springer*, vol. 387, pp. 305-312, 2016, doi: 10.1007/978-3-319-32213-1_27.
- [10] T. Ahmed, "Fitness Dependent Optimizer: Inspired by the Bee Swarming Reproductive Process," in *IEEE Access*, vol. 7, pp. 43473-43486, 2019, doi: 10.1109/ACCESS.2019.2907012.
- [11] Z. Gao, Y. Xu, F. Meng, F. Qi and Z. Lin, "Improved information gain based feature selection for text categorization," in *Proc. ICWC, Chennai, India*, pp. 1-5, 2011.
- [12] T. Pham, E. Foo, S. Suriadi and H. Jeffrey, "Improving performance of intrusion detection system using ensemble methods and feature selection," *Australasian Computer Science Week Multi-Conference, ACM*, vol. 2, pp.1-6, 2018, doi: 10.1145/3167918.3167951.
- [13] L. Dhanabal and S.P. Shantharadah, "A study on NSLKDD dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol.4, no.6, pp. 446-452, 2015.
- [14] C. Yin, Y. Zhu, J. Fei and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," in *IEEE Access*, vol. 5, pp. 21954-21961, 2017, doi: 10.1109/ACCESS.2017.2762418.
- [15] K. M. Sydne and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *Journal of Big Data*, vol. 7, no. 1, 1-20, 2020, doi: 10.1186/s40537-020-00379-6.
- [16] M. Hammad, W. El-medany and Y. Ismail, "Intrusion Detection System using Feature Selection With Clustering and Classification Machine Learning Algorithms on the UNSW-NB15 dataset," *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, 2020, pp. 1-6, doi: 10.1109/3ICT51146.2020.9312002.
- [17] Neyole Misiko Jacob and Muchelule Yusuf Wanjala, "A Review of Intrusion Detection Systems," *Global Journal of Computer Science and Technology: C Software & Data Engineering*, vol. 17, no. 3, 2017.
- [18] A. Kharaisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 18, pp. 1-27, 2021, doi: 10.1186/s42400-021-00077-7.
- [19] A. Chauhan, R. Singh, and P. Jain, "A Literature Review: Intrusion Detection Systems in Internet of Things," *Journal of Physics Conference Series*, vol. 1518, p. 012040, 2020, doi: 10.1088/1742-6596/1518/1/012040.
- [20] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255-277, 2017, doi: 10.1016/j.cose.2017.06.005.
- [21] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert Systems with Applications*, vol. 148, p. 113249, 2020, doi: 10.1016/j.eswa.2020.113249.
- [22] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," *Symmetry*, vol. 12, no. 6, p. 1046, 2020, doi: 10.3390/sym12061046.
- [23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *computers & security*, vol. 82, no. 1, pp. 35-45, 1960, doi: 10.1115/1.3662552.
- [24] A. T. phan, G. Herman, P. Wira, "The Discrete Kalman Filter, State-Space Modeling and Simulation," in *2015 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, Canada, 1997, vol. 2, pp. 190-241, 2015, doi: 10.1109/EAIS.2015.7368807.
- [25] P. Zarchan and H. Musoff, "Fundamentals of Kalman Filtering: A Practical Approach," *2nd ed.; AIAA: Alexandria*, 2015.

BIOGRAPHIES OF AUTHORS



Mathiyalagan Ramasamy     received the Bachelor of Technology in Information Technology in 2005 from Dr. Navalar Nedunchezian college of Engineering, Affiliated by Anna University, Tamil Nadu, India, and He received the Master of Engineering in Computer Science and Engineering in 2010 from Oxford college of Engineering, Affiliated by Anna University, Tamil Nadu, India, He is Assistant professor, Department of information science and engineering, Faculty of Engineering and Technology-Jain (Deemed to-be University). His research interests include intrusion detection system, convolutional neural network, and network security. He can be contacted at email: mathi.prajval@gmail.com.



Dr. Pamela Vinitha Eric     is a Professor at the Department of Computer Science and Engineering, New Horizon college of Engineering, Bengaluru, India, where she has been a faculty member since 2018. From 2013-2018 and worked in Rajiv Gandhi Institute of Technology-Bangalore (Affiliated to VTU) as Associate Professor and Head of Information Science and Engineering Department. Sri Venkateshwara college of Engineering-Bangalore (Affiliated to VTU) as Assistant Professor in the Department of Computer Science and Engineering. She can be contacted at email: pamela.vinitha@gmail.com.